Machine Learning with Graphs (MLG)

Final Project Report

吳岱桀 E14051350

1 INTRODUCTION

Anomaly detection is an important task in machine learning, which is a task of identifying anomalies in a dataset. Anomalies are a type of data that are rare but it would bring about greatly harmful effect on the overall performance. For example, the type of cancer instances is relatively rare compared to the normal instances in clinical data so it would often cause extremely data imbalance challenge. Other than medical domain, anomaly detection methods have been used in a wide variety of security-related applications such as financial fraud detection, social spam detection.

Although many anomaly detection methods have been proposed such as one-class support vector machines (OSVM), autoencoder (AE), and isolation forest, but there's little attention specifically focused on dealing anomaly detection with missing values. In many fields including biology, clinical, finance, economics and manufacturing industry, incomplete data tent to become a certain problem and the situation would become even worser if there's anomaly existing. The missing data problem has previously been formulated as label prediction task aiming at directly complete object of classification or regression with the missing values present in the input data.

In this project, we propose to deal with the anomaly detection task with existing missing value by the means of label prediction. Formulating the problem using a graph representation and applying Graph Neural Network(GNN), where we construct a bipartite graph with observations and features in tabular data as two types of nodes, and the observed feature values as attributed edges between the observation and feature nodes. Under this graph representation, the label anomaly prediction is treated as a node-level prediction task to classify whether a certain observation is normal or anomaly.

Most existing anomaly detection methods on attributed graphs are done in unsupervised learning that they do not consider the label (normal and anomalous) information of nodes. In fact, although label information is often rare in anomaly detection application, it may have potential advantage on the other hand. In order to utilize the valuable label information, semi-supervised learning methods for an attributed graph can use this label information to classify unlabeled instances. In this project, we would also try to make good use of label information to make model have a better decision boundary.

2 METHODOLOGY

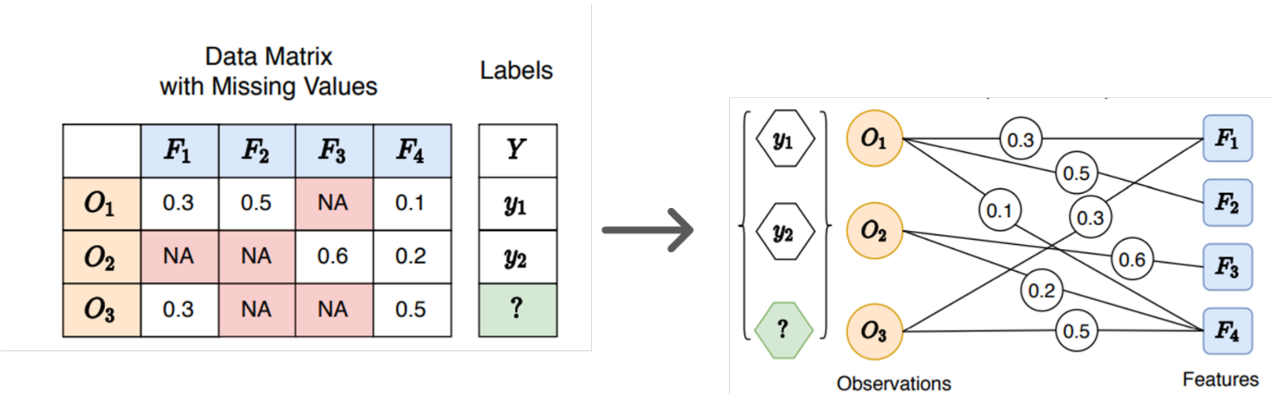
2.1 Problem Formulation

The target of this project is to predict whether a observation in the dataset is a normal or anomaly instance under the condition with existing missing value. In other words, we would need to solve imbalanced classification problem and handle the missing value problem at the same time.

2.2 Previous Work

Although there're research to discuss anomaly detection in in attributed graph, but there's very few research about solving anomaly detection and missing value task by formulating it to graph domain problem. In this section, we would introduce related research and how we modify those works that make it appropriate to be utilized in our project.

2.2.1 Missing Value



Data Matrix with Missing Values

	F_1	F_2	F_3	F_4
O_1	0.3	0.5	NA	0.1
O_2	NA	NA	0.6	0.2
O_3	0.3	NA	NA	0.5

Labels
Y
y_1
y_2
?

In order to convert the missing value problem to graph domain, we adopt the work in [1]. As illustrated in the above figure, the method to transform the missing value task to graph is to represent the feature matrix with missing values as a bipartite graph, which each of observation would be treated a type of node while feature would be the other type of node. If it is a discrete variable then it is transformed to a one-hot vector. The weighted edge(link) between observation node and feature node is the value of feature. Then the anomaly detection task would be label prediction problem that can naturally be formulated as node prediction issue. For example, y_1 and y_2 are known label indicating it's normal or anomaly observation for o_1 and o_2, the anomaly detection problem with missing value is to predict the node label y_3 for o_3.

Node prediction for graph can be further divided into two settings based on the way how new data is handled: (1) transductive setting and (2) inductive setting. The former performs node prediction on a single and fixed graph that includes those nodes we want to know, while the latter attempts to handle newly observed nodes or graphs that do not appear in the training process with a previously learned model. To achieve inductive setting in GNN, setting of node feature and the process to generate node embedding by applying aggregation function should be both taken care of.

The node feature for observation node would be initialized as m-dimensional constant vectors and for feature nod would be m-dimensional one-hot node features for each node (m is the number of total features).

$$\textsc{Init}(v) = \begin{cases} \mathbf{1} & v \in \mathcal{V}_D \\ \textsc{OneHot} & v \in \mathcal{V}_F \end{cases}$$

$$\mathbf{n}_v^{(l)} = \textsc{Agg}_l\Big(\sigma(\mathbf{P}^{(l)} \cdot \textsc{Concat}(\mathbf{h}_v^{(l-1)}, \mathbf{e}_{uv}^{(l-1)}) \mid \forall u \in \mathcal{N}(v, \mathcal{E}_{drop}))\Big)$$

GAT

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

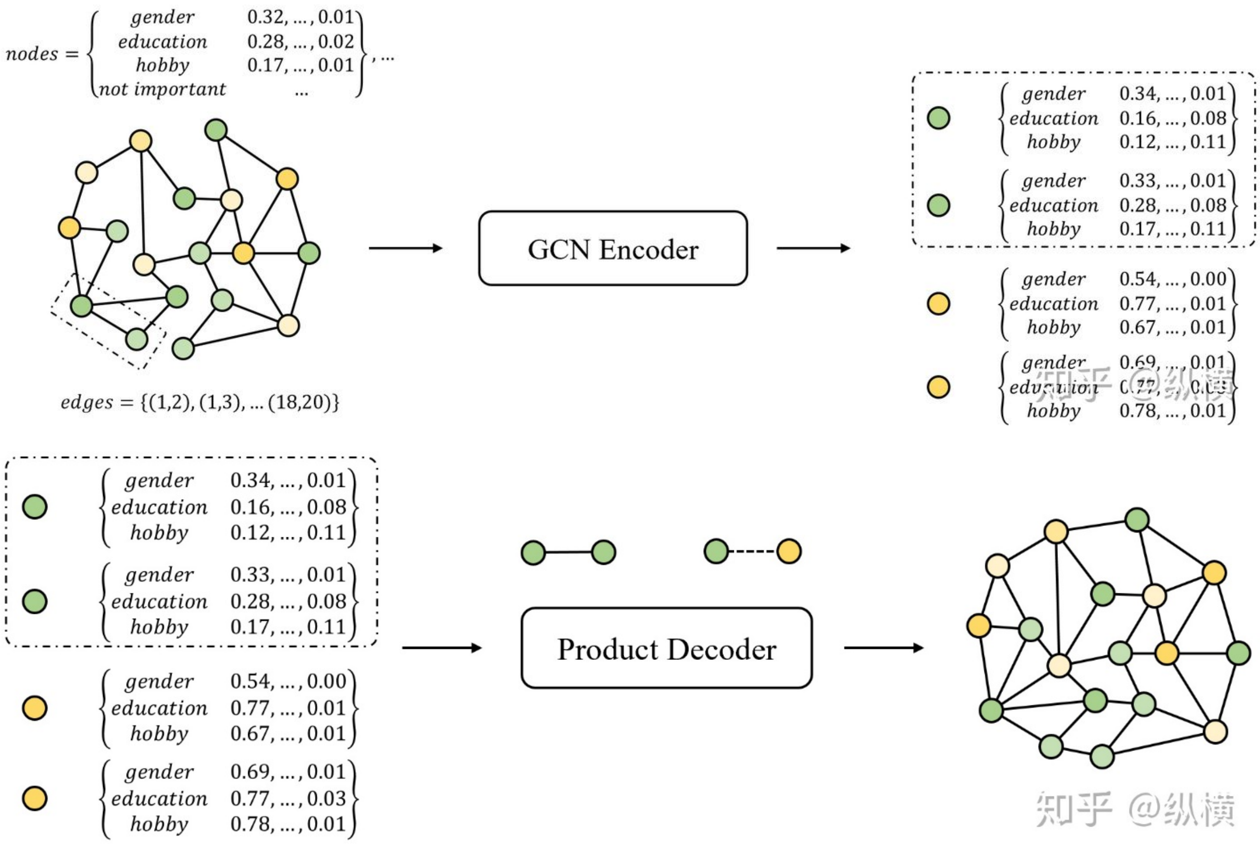
$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right)$$

$$\mathbf{h}_v^{(l)} = \sigma(\mathbf{Q}^{(l)} \cdot \textsc{Concat}(\mathbf{h}_v^{(l-1)}, \mathbf{n}_v^{(l)}))$$

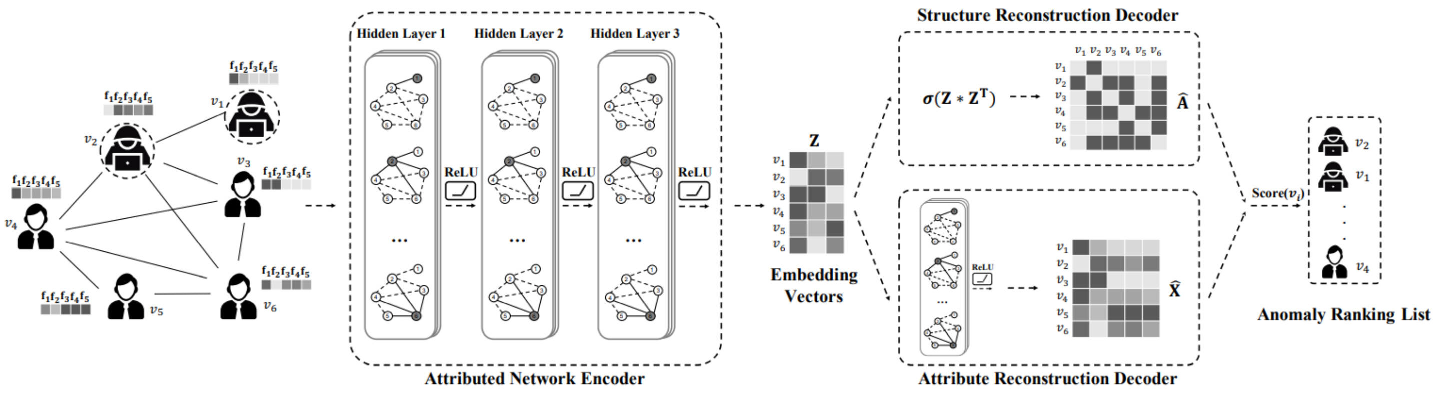
$$\mathbf{e}_{uv}^{(l)} = \sigma(\mathbf{W}^{(l)} \cdot \textsc{Concat}(\mathbf{e}_{uv}^{(l-1)}, \mathbf{h}_u^{(l)}, \mathbf{h}_v^{(l)}))$$

To fulfill the condition of inductive learning to generate node embedding, it’s necessary that the aggregating process should not use any structure information such as node degree in the graph. Combined with [1] and the idea of attention mechanism in Graph Attention Network [2], we propose to use the following process to obtain embedding of each node where the aggregation function would adopt the multi-head attention method in GAT to achieve inductive learning. A special innovation in [1] that’s worth noticed is that they propose the concept using edge embedding on weighted graph that allows it to get a better representation of node embedding.

2.2.2 Anomaly Detection in Graph Domain


nodes = {gender 0.32, ..., 0.01; education 0.28, ..., 0.02; hobby 0.17, ..., 0.01; not important ...}, ...
edges = {(1,2), (1,3), ... (18,20)}
GCN Encoder
gender 0.34, ..., 0.01
education 0.16, ..., 0.08
hobby 0.12, ..., 0.11
gender 0.33, ..., 0.01
education 0.28, ..., 0.08
hobby 0.17, ..., 0.11
gender 0.54, ..., 0.00
education 0.77, ..., 0.01
hobby 0.67, ..., 0.01
gender 0.69, ..., 0.01
education 0.77, ..., 0.03
hobby 0.78, ..., 0.01
Product Decoder

Most work for solving anomaly detection in graph domain follows the architecture of Graph Autoencoder (GAE)[4] as illustrated above. The main idea of those works to identify anomaly node is using encoder to extract node embedding and decoder to reconstruct the node feature and graph structure from the node embedding. If the information of the graph can be approximated through the structure reconstruction decoder, it implies that the probability to be anomalous is low.

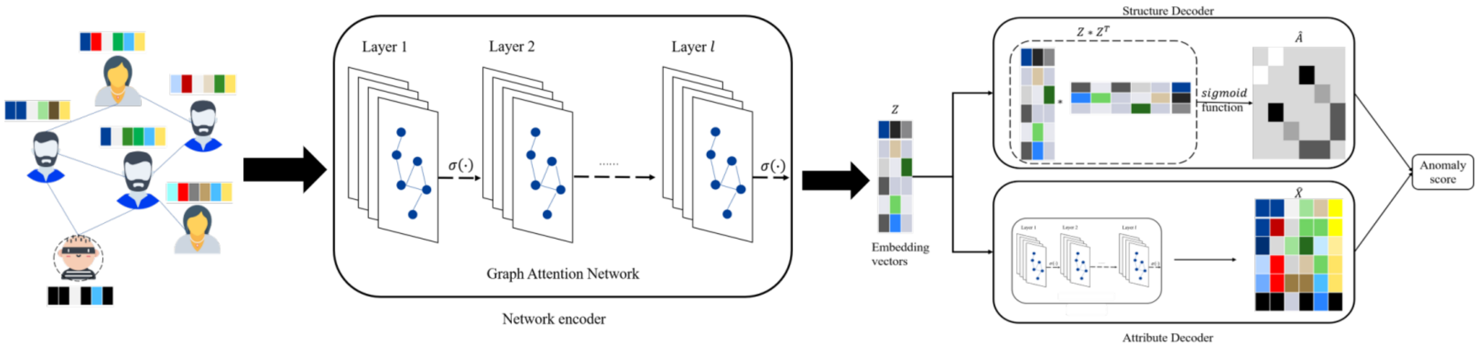


The above proposed Dominant model in [3] can spot anomalies by analyzing the reconstruction errors of nodes from both the structure and the node feature perspectives. It applies a 3-layer Graph Convolutional Network (GCN) [5] as the encoder in Dominant, while there are two types of decoder which are structure reconstruction decoder and attribute reconstruction decoder. The node embedding relationship between two layers of GCN can be expressed as the following formula:

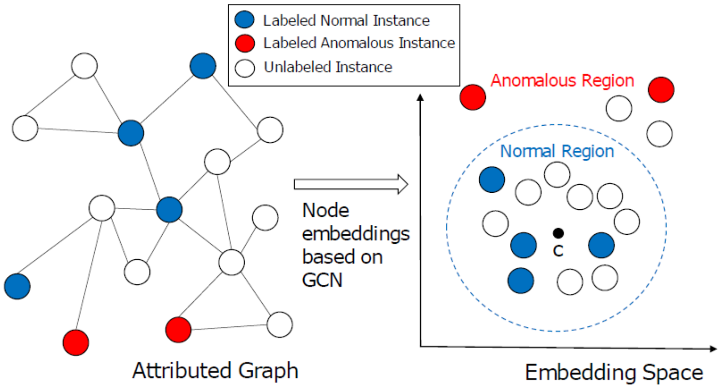
$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

H denotes the node embedding at different layer. D is degree matrix of nodes, while W is learnable weights.

Structure reconstruction decoder aims to reconstruct the original graph topology with the learned node embeddings and attribute reconstruction decoder attempts to reconstruct the observed node feature with the obtained node embeddings.Structure reconstruction decoder utilize inner-product operation on the node embedding and the main goal of it is to accomplish the link prediction in order to minimize the reconstruction error between reconstructed adjacency matrix and the true one. It suggests that if the connectivity patterns cannot be well reconstructed, its structure information does not conform to the patterns of majority normal nodes. The attribute reconstruction decoder is using a single layer GCN to reconstruct the node feature from the node embedding.



The GATAE model in [6] is a refined architecture of Dominant. As shown in figure above, it improves the overall performance by adopting 3-layer GAT to obtain a better representation of node embedding. Besides, different from Dominant that uses a single layer GCN as the attribute reconstruction decoder, GATAE suggests taking 3-layer GAT to reconstruct node feature. The significant difference between GATAE and Dominant is that GATAE can be applied in inductive learning application.


● Labeled Normal Instance
● Labeled Anomalous Instance
○ Unlabeled Instance
Anomalous Region
Normal Region
c
Node embeddings based on GCN
Attributed Graph
Embedding Space

Anomaly Score: $a(v_n) := \|\mathbf{h}_n - \mathbf{c}\|^2$

The proposed model in [7] is taking different strategy from the scope of GAE that they introduce the concept of one-class classification and the idea of unsupervised-learning. As demonstrated in above figure, the parameters of a GCN are trained to minimize the volume of a hypersphere that encloses the node embeddings of normal instances while embedding anomalous ones outside the hypersphere. The anomaly score defined above can be estimated to identify anomaly instance, where h_n is node embedding of normal instance from GCN and c is hypersphere center c by calculating mean of the node embedding for normal instance after performing an initial forward pass. The loss function in [7] is defined as following formulas, which minimize the anomaly score for normal instance and take small amount of labeled anomaly instance into consideration by applying differential approximation of AUC.

Hypersphere Loss(using normal only)

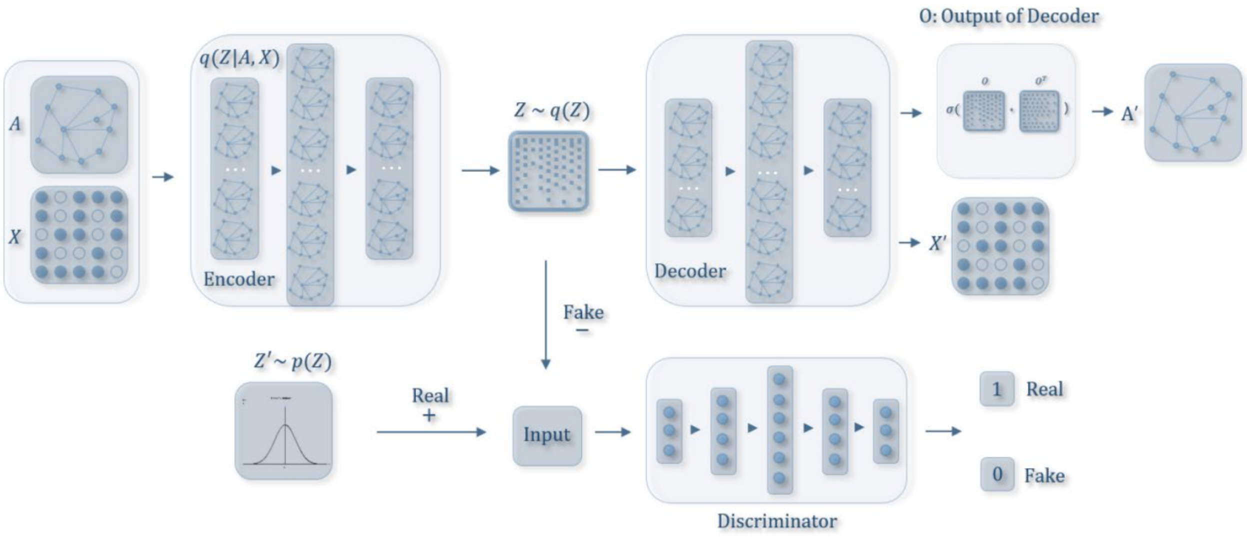
$$\mathcal{L}_{\text{nor}}(\theta) := \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} a(v_n) = \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} \|\mathbf{h}_n - \mathbf{c}\|^2$$

Differential Approximation of AUC

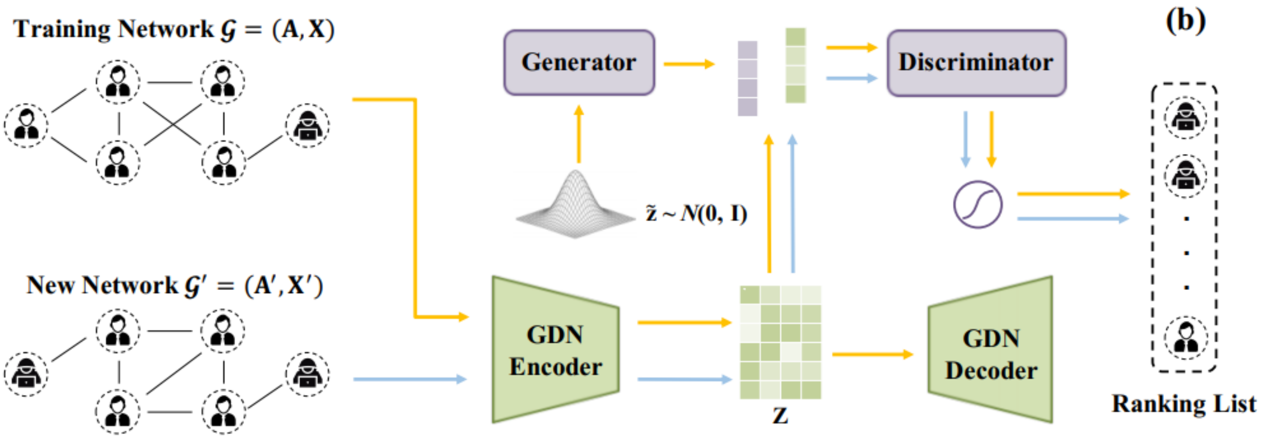
$$\mathcal{R}_{\text{AUC}}(\theta) := \frac{1}{|\mathcal{A}||\mathcal{N}|}\sum_{n\in\mathcal{A}}\sum_{m\in\mathcal{N}} f(a(v_n) - a(v_m))$$

Total Loss

$$\mathcal{L}(\theta) := \mathcal{L}_{\text{nor}}(\theta) - \lambda\mathcal{R}_{\text{AUC}}(\theta)$$



As the success by employing Generative Adversarial Networks (GAN) [8] to solve anomaly detection problem, it proves that GAN can effectively improve the ability of model from distribution aspect. Inspired by previous work, [9] utilizes GCN as an encoder that embeds the topological information and node content into a vector representation, from which a graph decoder is further built to reconstruct the input graph. The adversarial training principle is applied to enforce our latent codes to match a prior Gaussian or uniform distribution as illustrated in the above graph.


Training Network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$
New Network $\mathcal{G}' = (\mathbf{A}', \mathbf{X}')$
Generator
Discriminator
(b)
$\tilde{\mathbf{z}} \sim N(\mathbf{0}, \mathbf{I})$
GDN Encoder
Z
GDN Decoder
Ranking List

AEGIS [10] formulates anomaly detection with GAN in a different way. The main goal to introduce GAN in [9] is to restrict the node embedding to follow a certain distribution, while GAN used in AEGIS helps the model to learn a better classifier(discriminator) that trains generator to produce simulated node embedding. AEGIS are jointly trained in two phases and each phase requires dedicated training objective functions. The first stage aims at minimizing the reconstruction error in a GAE framework. After the training of first stage finishes, it then adopt the node embedding obtained from the encoder as ‘real’ reference and train the generator of

GAN to get a node embedding that is real enough to fool the discriminator.

2.3 Proposed Method

In this project, we would compare the different setting of the model to achieve inductive anomaly detection with missing value. We would adopt the work in [1] to transform tabular data into graph to get node embedding. Afterwards, the method discussed in the previous section would be incorporated to identify anomalous data from the node embedding.

3 EXPERIMENT & ANALYSIS

In this section, the dataset and how we evaluate the performance on different setting are shown in 3.1.1 and 3.1.2. The result of the experiment is summarized in 3.1.3.

3.1.1 Datasets

Arrhythmia: A cardiology dataset from the UCI repository containing attributes related to the diagnosis of cardiac arrhythmia in patients. The datasets consists of 16 classes: class 1 are normal patients, 2-15 contain different arrhythmia conditions, and class 16 contains undiagnosed cases. Following the protocol established in previous works, the smallest classes: 3; 4; 5; 7; 8; 9; 14; 15 are taken to be anomalous and the rest normal.

Thyroid: A medical dataset from the UCI repository , containing attributes related to whether a patient is hyperthyroid. We designate hyperfunction as the anomalous class and the rest as normal from the 3 classes of the dataset.

KDD: The KDD Intrusion Detection dataset was created by an extensive simulation of a US Airforce LAN network. The dataset consists of the normal and 4 simulated attack types: denial of service, unauthorized access from a remote machine, unauthorized access from local superuser and probing.The dataset consists of around 5 million TCP connection records. We use the UCI KDD 10% dataset, which is a subsampled version of the original dataset. The dataset contains 41 different attributes. 34 are continuous and 7 are categorical.

KDDCUP99-Rev: To better correspond to the actual use-case, in which the non-attack scenario is normal and attacks are anomalous, reverse configuration are evaluated, in which the attack data is subsampled to consist of 25% of the number of non-attack samples. The attack data is in this case designated as anomalous (the reverse of the KDDCUP99 dataset).

3.2 Performance Evaluation

To measure the performance of our trained model, we utilize F1 score. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account that make it a good measure for the imbalanced dataset.

3.3 Result

All the methods we introduced in this project adopt same procedure to generate node embedding as the one in [1]. The major difference in these methods is the way to utilize the node embedding for anomaly detection problem.

Classification: After node embedding is obtained, using a single linear layer to identify whether a certain node is normal or anomaly from the node embedding.

Reconstruction Error: The inner-product decoder is used to reconstructed adjacency matrix from node embedding. The attribute decoder in previous works is not adopted in this project because that the node feature contains relatively little information, which observation node feature is constant vector and one-hot feature for the feature node. We believe that it's more important to focus on reconstructing the graph structure. To identify anomalous data, we empirically set up a threshold that those observation whose reconstruction higher than threshold would be classified as anomaly.

Semi-supervised: We employ it as the same setting in [7].

Semi-supervised + GAN: Combined the idea of [7] and [9], GAN is used to restrict node embedding follows a hypersphere distribution.

Semi-supervised + 2-stage GAN: Combined the idea of [7] and [10], GAN is used to train a better classifier.

The result without missing value is presented as baseline. In addition to the methods we've conducted in this project, 3 machine learning methods specific for anomaly detection are shown in this section as baseline.

Method	Dataset			
	Arrhythmia	Thyroid	KDD	KDDRev
OC-SVM	45.8	38	79.5	83.2
LOF	50	52.7	83.8	81.6
Iforest	51.4	63	90.7	87.7
Classification	51.7	69.7	92.4	93.5
Reconstruction Error	51.3	65	91.3	93
Semi-supervised	58.2	79.6	96.5	97.2
Semi-supervised + GAN	60.5	80.5	97.9	98.5
Semi-supervised + 2-stage GAN	59.2	80.1	97.1	98

To compare the result when missing value is appeared, we randomly choose 10% value of feature to become unobserved. 3 machine learning methods use MICE to complete imputation.

Method	Dataset			
	Arrhythmia	Thyroid	KDD	KDDRev
OC-SVM	38.1	33.6	72.1	76.9
LOF	40.5	48.7	76.5	73.2
Iforest	42	57.2	82.9	79.1
Classification	46.8	62.2	84.6	85.1
Reconstruction Error	44.5	60.9	83.5	84.5
Semi-supervised	50.3	73.8	88.2	90.3
Semi-supervised + GAN	52.9	74.7	89.7	91
Semi-supervised + 2-stage GAN	51.7	74.1	89.9	90.4

As shown above, the basic operation like Classification and Reconstruction Error has slight improvement compared to baseline, which means it can extract useful information from graph structure. By using semi-supervised method, the performance has a more significant improvement, and it proved to be effective with the help of GAN.

4 Conclusion

In this project, we have successfully formulate anomaly detection with the problem of missing value as node prediction task. By the combination from both concept of semi-supervised learning and idea of GAN, the method proposed indeed outperform previous machine learning methods in anomaly detection problem either with or without missing value .

5 REFERENCE

[1] You, J., Ma, X., Ding, D. Y., Kochenderfer, M., & Leskovec, J. (2020). Handling missing data with graph representation learning.

[2] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks.

[3] Ding, K., Li, J., Bhanushali, R., & Liu, H. (2019, May). Deep anomaly detection on attributed networks. In Proceedings of the 2019 SIAM International Conference on Data Mining (pp. 594-602). Society for Industrial and Applied Mathematics.

[4] Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders.

[5] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks

[6] You, Z., Gan, X., Fu, L., & Wang, Z. (2020, August). GATAE: Graph Attention-based Anomaly Detection on Attributed Networks. In 2020 IEEE/CIC International Conference on Communications in China (ICCC) (pp. 389-394). IEEE.

[7] Kumagai, A., Iwata, T., & Fujiwara, Y. (2020). Semi-supervised Anomaly Detection on Attributed Graphs.

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[9] Pan, S., Hu, R., Fung, S. F., Long, G., Jiang, J., & Zhang, C. (2019). Learning graph embedding with adversarial training methods. IEEE transactions on cybernetics, 50(6), 2475-2487.

[10] Ding, K., Li, J., Agarwal, N., & Liu, H. (2020). Inductive anomaly detection on attributed networks. In 29th International Joint Conference on Artificial Intelligence, IJCAI 2020 (pp. 1288-1294). International Joint Conferences on Artificial Intelligence.